

# 第3章：コンテナデータ型とその操作

春山 鉄源

2025/10/13, 10:22

コンテナデータ型とは、任意の数のデータ型の値を格納するデータ型のことです。格納された値にアクセスしたり、それらを反復処理をしたりすることができます。

## 1 4つのコンテナデータ型

ここでは次の4つについて考えます。\* リスト (list) \* タプル (tuple) \* 辞書 (dict) \* 集合 (set)

### 1.1 リストとタプル

- リスト：角括弧 [] で作成
  - 要素が複数ある場合は各要素はコンマ, で区切る
  - 変更可能
- タプル：丸括弧 () で作成
  - 要素が複数ある場合は各要素はコンマ, で区切る
  - 変更不可

コード 3.1.1

```
lst = [10, True, "国内総生産"]
```

コード 3.1.2

```
tpl = (10, True, "国内総生産")
```

コード 3.1.3

```
tpl = 10, True, "国内総生産"
```

コード 3.1.4

```
t0, t1, t2 = 10, True, "国内総生産"
```

タプルのアンパッキング (unpacking)

10 → t0,      True → t1,      "国内総生産" → t2

## 1.2 辞書と集合

- 辞書：波括弧{}で作成
  - キーと値のペアで一つの要素
  - コロン: の左がキー、右が値

{キー: 値}

  - 要素が複数ある場合は各要素はコンマ, で区切る
  - 変更可能
- 集合：波括弧{}で作成
  - 要素が複数ある場合は各要素はコンマ, で区切る
  - 要素が一つの場合もコンマ, が必要

(要素1,)

  - 変更可能

コード 3.1.5

```
# CELL PROVIDED

{10, 10, 10, True, True, True, 4, "Kobe", "Kobe"}
```

9つの要素がありますが、コードを実行すると、重複する値は削除され4つの要素の集合となっています。この重複要素削除は、後のシミュレーションで使います。

コード 3.1.6

```
# CELL PROVIDED

dct = {"GDP":557, "消費":298, "投資":111}
```

## 2 コンテナデータ型の操作

### 2.1 リストとタプル：要素のアクセス

コード 3.2.1

```
# CELL PROVIDED

abc = ["A", "B", "C", "D", "E", "F"]
```

0	1	2	3	4	5	(最初から数える)
-----						
["A", "B", "C", "D", "E", "F"]						
-----						
-6	-5	-4	-3	-2	-1	(最後から数える)

ここで、0, 1, 2、や-1、-2をインデックス (index) と呼ぶ。  
要素のアクセス方法：

角括弧 [] とインデックスを使う。

コード 3.2.2

```
abc[2]
```

コード 3.2.3

```
# CELL PROVIDED
```

```
lst0 = [{"GDP", 557}, {"消費", 298}, {"投資", 111}]
```

```
lst0[-1][0]
```

コード 3.2.4

```
lst0_last = lst0[-1]
```

```
lst0_last[0]
```

コード 3.2.5

```
# CELL PROVIDED
```

```
tpl = (10, True, "国内総生産")
```

```
tpl[1]
```

## 2.2 リストとタプル：要素のスライシング

連続する複数要素を選択する場合（スライシング）は:（コロン）を使う。

: の左側が選択する最初の要素のインデックス

: の右側が選択する最後の要素の次のインデックス

: の左側のインデックスを省略すると「最初から」となる

: の右側のインデックスを省略すると「最後まで」となる

例 3.2(a)

```
abc[1:4]
```

例 3.2(b)

```
abc[1:3+1]
```

例 3.2(c)

```
abc[:3]
```

例 3.2(d)

```
abc[3:]
```

例 3.2(e)

```
abc[3:-1]
```

## 2.3 辞書

例 3.3(a)

```
dct["GDP"]
```

例 3.3(b)

```
dct["消費"]
```

例 3.3(c)

```
dct["投資"]
```

## 2.4 要素の更新

コード 3.2.6

```
abc[2] = 100  
abc
```

コード 3.2.7

```
dct["投資"] = 200  
dct
```

## 2.5 要素の削除

授業では扱わない

## 2.6 要素の追加

コード 3.2.10

```
lst1 = ["消費", "投資"]  
lst1.append("政府支出")  
lst1
```

コード 3.2.11

```
dct["政府支出"] = 145
dct
```

### 3 関連トピック

#### 3.1 リストとタプル：連結

授業では扱わない

#### 3.2 辞書：結合

授業では扱わない

#### 3.3 文字列からの文字抽出

授業では扱わない

#### 3.4 改行

コード 3.3.7

```
# CELL PROVIDED

long_variable_name_1 = 10
long_variable_name_2 = 20

(
    long_variable_name_1 +
    long_variable_name_2
)
```

コード 3.3.8

```
# CELL PROVIDED

my_lst = ["消費",
          "投資",
          "政府支出"]
```

コード 3.3.9

```
# CELL PROVIDED

my_dct = {"消費":298,
          "投資":113,
          "a":[1,2]}
```

### 3.5 所属演算子

授業では扱わない